REMARKS

Applicant has reviewed the Office Action dated October 9, 2007, and the reference cited therein. Applicant has amended the specification, claims and abstract to address informalities cited in the Office Action. **Claims 1-8** were previously pending and none were allowed. Claim 1 has been amended to include the elements previously recited in claim 4, and claim 8 has been amended to recite elements that parallel the recited elements of claim 1. New method claims 9-13 parallel the elements of apparatus claims 2-3 and 5-7. It is believed that the claim amendments overcome the claim rejections. Support for the amendments to claims 1 and 8 can be found at page 4, lines 3-5 and page 6, lines 13-17.

With reference to items 4 and 5 of the Office Action, Applicant replaced "111" by "215" at page 8, line 19. Thus, the objection relating to the drawings is overcome.

In view of the amendments, Applicant submits that the presently pending claims are patentable over the presently known prior art. Accordingly, Applicant requests favorable reconsideration of the previous objections and rejections in view of Applicant's amendments.

Please charge any fee deficiencies to Deposit Account No. 12-1216.

Grounds for Rejecting the Claims

The following identifies the authority and prior art applied to the identified claims for each rejection of the claims set forth in the Office Action.

1. Claims 1-6 and 8 are rejected under 35 U.S.C. §102(b) as anticipated by Karp U.S. Pat. No. 5,748,936 (Karp).
2. Claim 7 is rejected under 35 U.S.C. §103(a) over Karp.

Applicant traverses each of the grounds for the rejection of the presently pending claims for the reasons set forth herein below.

Reasons for Traversing the Current Rejections of the Claims

Regarding the general claimed subject-matter of the claims, in the case of *time-stationary* coding, every instruction that is part of the processor's instruction-set controls a complete set of operations that have to be executed in a single machine cycle. These operations may be applied to several different data items traversing the data pipeline. In this case it is the responsibility of the programmer or compiler to set up and maintain the data pipeline. The resulting pipeline schedule is fully visible in the machine code program.

In contrast to the above described time-stationary coding, in the case of data-stationary encoding, every instruction that is part of the processor's instruction-set controls a complete sequence of operations that have to be executed on a specific data item, as the data item traverses the data pipeline. Once the instruction has been fetched from program memory and decoded, the processor controller hardware ensures that the composed operations are executed in the correct machine cycle.

The claimed invention relates to a time-stationary processor, which is illustrated in Figures 1 and 2, in that a controller (CTR), that decodes the instructions, directly controls addressing of register files RF1, RF2 with signals WR1, WR2, RR1, RR2 and controls selecting the data path with signals WS1, WS2. Time-stationary encoding is often used in application-specific processors since it saves the overhead of hardware necessary for delaying the control information present in the instructions, at the expense of larger code size.

Time-stationary processors cannot support conditional operations, i. e. operations that return a result based on a condition computed at run-time. Time-stationary encoding demands that all control information, including the write back of results to a register file, is statically determined at compile time and encoded in the program. However, according to the invention recited in claim 1, operations to be executed by an execution unit that potentially produce more than one valid output are allowed. Based on the outcome of the execution, one of the outputs is selected to be written to a particular register file.

By way of example, the processor may handle a conditional instruction MODULO(A,C). This operation is often used in address calculations. The calculation is carried out for values of $C > 0$ and for values of A in the range between $-2C$ and $2C-1$. The result R of the MODULO(A,C) operation can be defined in c-code as follows:

```
If(A-C)>0 then R = A-C

If(A+C)<0 then R = A+C
```

For the above-mentioned range of values of $C > 0$ the conditional results are mutually exclusive.

In the claimed time-stationary processor embodying the present invention, the MODULO operation described above can be implemented on a combination of a first and a second functional unit. Therein the first functional unit calculates the function POSDIFF(A,C) as R = A - C and provides an identifier on the validity of the relation (A-C) > 0. The second functional unit calculates the function NEGSUM(A,C) as R = A + C and provides an identifier on the validity of the relation (A+C) < 0. Based on the value of A and C either the outcome of the first functional unit, or the outcome of the second functional unit, or none of these outcomes is written into the register file.

The Rejection of Claims 1-6 and 8 as Anticipated by Karp

The claimed dynamic completion of a write operation within a time-stationary processor, illustratively shown in the above example, is neither disclosed nor even remotely suggested by Karp. It is first noted that Karp does not even disclose or suggest a time-stationary processor. As shown in Figure 2 of Karp, the instruction unit 34 that decodes the instructions merely controls the functional units 30. Control of the register files 32 is handled by the functional units 30. Also from the code samples in col. 8, Karp concerns a data-stationary processor.

The first example is:

```
1. p1?ld r7=r9
2. p1?cmpeq.u p4,p0=r7,0
3. p4?ld r2=(r10)
4. p4?ld r3=(r11)
5. p4?cmpeq.u p2,p3=r2,0
6. p2?ld r4=(r12)
7. p3?ld r5=(r13)
```

The second example, showing a speculative version of this code for a processor with a SLAT is as follows:

```
1. p1?ld r7=r9
2. p1?lblpred,11 p4
3. p4?ld r2=(r10)
4. p4?ld r3=(r11)
5. p4?cmpeq.u p2,p3=r2,0
```

6. p2?ld r4=(r12)
7. p3?ld r5=(r13)
8. p1?cmpeq p5,p0=r7,0
9. p5?chklbl,11,immed p4

In both of Karp's examples, the assembly code is provided with all of its operands, including predicate, destination registers and argument registers. Pipelining of the operation is not visible in this code sequence. Accordingly, all information necessary to process a data value is atomically encoded with the operation, which indicates that Karp relates to *data-stationary* processing – not the claimed *time-stationary* processor.

Karp furthermore does not disclose how writing result data can be dynamically controlled on the basis of validity of an output result. Karp merely discloses how exceptions should be handled in the case of speculative execution. To that end Karp has a complex arrangement including a speculative look aside table, a separate file for storing predicates and a label decoder. Karp's mechanism is not suitable for allowing operations to be executed by the execution units that potentially produce more than one valid output. Either it is decided that no exception has occurred and consequently the output is accepted or it is decided that an exception has occurred and the output is ignored.

Applicant's claimed dynamic write control based upon validity of an output result differs from Karp's disclosed speculative execution. In Karp's disclosed speculative execution, an operation is moved across a conditional branch that controls its execution. In that case the operation is executed but a condition independent from the outcome of the operation determines whether the result is used or not. Said condition is in the original instruction sequence already available before the operation is performed. In the case of Applicant's claimed dynamic write control on the basis of a validity of an output result, the validity of the output result can only be determined during the calculation of the output result, even during non-speculative execution.

Thus, in summary, the invention recited in presently pending independent claims 1 and 8 discloses a novel and inventive way to enable conditional dynamic write operations in a time-stationary processor. The claimed invention enables custom operations to be executed by the execution unit that potentially produce more than one valid output.

Applicant notes that, based upon the description of an illustrative example at page 3, lines 21-27, the control on the basis of the first identifier is optional. Accordingly the dynamic control of the transfer of result data may be based solely on the second identifier alone.

Applicant's above discussion implicitly addresses the previous rejection of claim 4 as anticipated by Karp. Applicant traverses the rejection of each of the remaining dependent claims as anticipated or obvious over Karp since each of the claims recites all of the limitations of independent claim 1. Applicant reserves the right to address the basis for the rejection of the dependent claims, if needed, at a later time in response to any changed grounds for the rejection of claim 1.

With reference to item 19 of the Office Action, Applicant submits that Branigin (US5471593) merely discloses increasing performance of a pipelined processor by executing instructions, conditional instruction execution issues and executes instructions, including but not limited to branches, before the controlling conditions may be available and makes the decision to update the destination as late as possible in the pipeline. Accordingly Branigin merely discloses a method of speculative execution, but does not disclose nor suggest dynamic control on the basis of a validity of an output result in a time-stationary processor.

Applicant respectfully submits that the patent application is in condition for allowance. If, in the opinion of the Examiner, a telephone conference would expedite the prosecution of the subject application, the Examiner is invited to call the undersigned attorney.

Respectfully submitted,

[illegible]

Mark Joy, Reg. No. 35,562
LEYDIG, VOIT & MAYER, LTD.
Two Prudential Plaza, Suite 4900
180 North Stetson Avenue
Chicago, Illinois 60601-6731
(312) 616-5600 (telephone)
(312) 616-5700 (facsimile)

Date: February 11, 2008